

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame – 07 Settembre 2022 – Versione A**

**Tempo a disposizione: 180 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>A1.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>A2.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 2
<b>A3.zip</b>	file zip contenente il sorgente javascript e pagine Web per punto 3

**Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto.**

**N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.**

---

**ESERCIZIO 1 (11 punti)**

Si realizzi una applicazione Web per **la scrittura collaborativa di un articolo scientifico da parte di più ricercatori**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere a un **utente non autenticato** di **inserire** una **stringa di al max 64 caratteri** che rappresenta il **nome dell'articolo**, **seguita dal carattere speciale '%'**: alla pressione del carattere speciale, **la stringa deve essere inviata automaticamente al server**; **se tale articolo è già stato creato, l'utente sarà associato a quell'articolo per tutta la sessione di interazione**; altrimenti un articolo con tale nome verrà creato vuoto lato server.

Una volta acceduto l'articolo **stringa**, **l'utente dovrà avere visualizzato l'attuale contenuto dell'articolo in una semplice finestra di testo, disponibile in sola lettura**; a fianco sarà presente un bottone per richiedere il permesso di scrittura sull'articolo. **Tale permesso viene concesso a un solo utente per volta per ogni articolo**; per semplicità, la scrittura avviene sempre in append al contenuto corrente. **Si faccia in modo che l'utente che possiede correntemente il permesso di scrittura abbia visualizzato anche un bottone per rilasciare tale permesso.**

In ogni momento, **previa autenticazione**, un **amministratore** dell'applicazione Web di scrittura collaborativa di articoli **deve avere la possibilità di visualizzare tutti gli articoli acceduti da almeno un utente e di revocare il permesso di scrittura** (ad esempio per evitare lock).

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame – 07 Settembre 2022 – Versione A**

**ESERCIZIO 2 (11 punti)**

Si realizzi una applicazione Web per l'**accesso veloce ed efficiente alle informazioni relative al tabellone degli US Open di tennis**. L'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, l'applicazione Web deve permettere a utenti non autenticati di inserire il **cognome** di un tennista partecipante agli US Open 2022 e di ottenere come risposta un testo descrittivo delle **statistiche** di quel tennista (ranking ATP, titoli vinti, numero partite vinte e perse nel 2022); nel caso di cognome non valido, deve essere visualizzata la frase "Non partecipa agli US Open 2022".

Visto che di solito gli utenti sono interessati alle statistiche di tennisti nella stessa parte del tabellone, per motivi di efficienza e velocità di risposta nelle richieste successive, si faccia in modo che ogni richiesta generi anche lo scaricamento in pre-fetching di tutte le informazioni relative ai tennisti che si trovano nella stessa parte di tabellone. A tal fine si consideri un tabellone composto da 128 tennisti, diviso in 4 parti, ciascuna costituita da 32 tennisti. Ad esempio, dopo avere richiesto le statistiche di "Musetti", che appartiene alla parte 2 del tabellone (ad esempio, tennisti con id numerico da 33 a 64), l'utente avrà già pre-fetched sul browser locale anche le info di tutti gli altri 31 tennisti della parte 2 del tabellone. In caso di richiesta futura su uno qualsiasi di quei 31 tennisti della parte 2, l'applicazione Web non deve generare alcun accesso al server Web ma semplicemente visualizzare le info già pre-fetched; ovviamente in caso di richiesta con cognome di tennista appartenente alle altre 3 parti di tabellone, si deve avere una invocazione "normale".

---

**ESERCIZIO 3 (11 punti)**

Si realizzi in React un'applicazione Web lato client che simuli il **gioco della "Caccia al tesoro"**. L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.

L'interfaccia dell'applicazione sarà composta dalle seguenti sezioni:

- **Sezione Configurazione.** In questa sezione sono presenti due elementi di input per l'inserimento delle dimensioni della griglia di ricerca (rispettivamente, larghezza e lunghezza). I campi larghezza e lunghezza non possono assumere valori inferiori a 5. All'interno di una cella di questa griglia, scelta in modo random, viene posizionato il tesoro;
- **Sezione Mappa del Tesoro.** Tale sezione deve contenere una griglia rettangolare (larghezza \* lunghezza) di celle di colore grigio. Il giocatore cerca di trovare il tesoro cliccando su una cella della griglia; effettuato il tentativo, la cella in questione si colorerà di giallo se non contiene il tesoro; diversamente, la cella si colorerà di blu e apparirà su di essa la lettera "T". Il gioco termina quando il giocatore avrà trovato il tesoro;
- **Sezione Punteggio.** In questa sezione, viene visualizzato in tempo reale il numero di tentativi infruttuosi da parte del giocatore. Quando il giocatore trova il tesoro, viene visualizzato anche il punteggio conseguito dal giocatore, così determinato: n. 5 punti se il tesoro viene trovato entro 10 tentativi, n. 2 punti se per trovare il tesoro sono stati necessari almeno 11 tentativi.